NASA-CR-199755

NAS2-13721
IN-64-CR
6386
p. 17

RIACS

Research Institute for Advanced Computer Science
NASA Ames Research Center

A Deterministic Particle Method for One-Dimensional Reaction-Diffusion Equations

Michael Mascagni

(NASA-CR-199755) A DETERMINISTIC PARTICLE METHOD FOR ONE-DIMENSIONAL REACTION-DIFFUSION EQUATIONS (Research Inst. for Advanced Computer Science) 17 p

N96-15853

Unclas

G3/64 0083947

RIACS Technical Report 95.23 November, 1995

A Deterministic Particle Method for One-Dimensional Reaction-Diffusion Equations

Michael Mascagni

The Research Institute of Advanced Computer Science is operated by Universities Space Research Association, The American City Building, Suite 212, Columbia, MD 21044, (410) 730-2656

Work reported herein was supported by NASA via Contract NAS 2-13721 between NASA and the Universities Space Research Association (USRA). Work was performed at the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center; Moffett Field, CA 94035-1000.

A DETERMINISTIC PARTICLE METHOD FOR ONE-DIMENSIONAL REACTION-DIFFUSION EQUATIONS

MICHAEL MASCAGNI

RIACS, NASA Ames Research Center
and
Center for Computing Sciences, I.D.A.

ABSTRACT. We derive a deterministic particle method for the solution of nonlinear reaction-diffusion equations in one spatial dimension. This deterministic method is an analog of a Monte Carlo method for the solution of these problems that has been previously investigated by the author. The deterministic method leads to the consideration of a system of ordinary differential equations for the positions of suitably defined particles. We then consider the time explicit and implicit methods for this system of ordinary differential equations and we study a Picard and Newton iteration for the solution of the implicit system. Next we solve numerically this system and study the discretization error both analytically and numerically. Numerical computation shows that this deterministic method is automatically adaptive to large gradients in the solution.

1. Introduction.

We are interested in numerical methods for the solution of nonlinear reaction-diffusion equations in one species and in one spatial dimension. These equations can be studied as initial-value problems and have the form:

$$u_t = \nu u_{xx} + f(u), \quad t > 0, \quad , u(x,0) = u_0(x), \quad \nu > 0. \tag{1}$$

Partial differential equations (PDEs) of this form are often found in multi-species systems that model phenomena from chemical combustion to excitability in neurons [10, 7, 11].

A qualitative feature of solutions to (1) is that they often have sharp gradients and traveling fronts. Because of this, standard finite-difference methods often require very fine grids to resolve the sharp gradients. With a traveling front solution the gradient is steepest in a small region that moves with the front. An optimal finite-difference method would incorporate moving refined grids into a numerical solution algorithm.

An alternative is to consider a Monte Carlo method for the solution of (1). Such an approach yields a particle-based method that is automatically adapting to sharp gradients, [18, 19, 20]. The major drawback with a Monte Carlo approach is that convergence is

1991 *Mathematics Subject Classification.* 65CM99, 92C20.

Key words and phrases. reaction-diffusion equations, particle methods, numerical analysis, automatic adaptivity.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

stochastic and can be shown to behave as $O(N^{-1/2})$, where N is the number of particles, [21]. An advantage of the Monte Carlo approach is that its particle-based formulation can serve as a starting point for the construction of a deterministic analog. This paper explores such a deterministic particle method for monotonic solutions to equation (1).

The plan of the paper is as follows. In §2 we briefly review a one-dimensional Monte Carlo method, called the gradient random walk (GRW) method, for (1) and recall it's major qualitative and quantitative properties. In §3 we derive the deterministic analog for monotonic solutions to equation (1). This leads to an interesting system of ordinary differential equations (ODEs) that has nonlinearities corresponding to the linear diffusion term in (1) and is linear where (1) is nonlinear! In §4 we discuss some common boundary conditions for reaction-diffusion equations and techniques for incorporating them into this numerical method. In §5 we consider the forward and backward Euler methods for the solution of the resulting ODEs. The backward Euler method leads to consideration of a nonlinear tridiagonal system. A Picard iteration is shown to solve this system; and a Newton's method solution is also analyzed. In §6 we present some numerical results for the Nagumo equation, a scalar reaction-diffusion equation with monotonic solutions that models nerve conduction. These numerical results motivate an analysis of the discretization error. This error can be further broken down into an error in the wave speed, that can be analyzed analytically, and a static discretization error, that can be studied numerically. Finally, in §7 we conclude by summarizing our results and discussing the extension of this method to multi-species reaction-diffusion systems and to higher dimensions.

2. Monte Carlo Algorithms.

We first review the gradient random walk (GRW) method [1, 5, 19] for equation (1). We assume that $u(\cdot,t)$ is monotonic, i.e., $u(x,t) < u(y,t)$ when $x < y$, and that $u(-\infty,t) = 0, u(+\infty,t) = 1$. The strategy is to represent $v = u_x$ by diffusing particles. We choose to use a particle representation for the gradient since integration of the gradient to recover the solution is a variance reduction technique. Differentiating (1) with respect to x,

$$v_t = \nu v_{xx} + f'(u)v,\ v(x,0) = u_0'(x), \tag{2}$$

and discretizing v as a sum of δ-functions with strength m_j, we have

$$v(x,t) = \sum_{j=1}^{N} m_j \delta\left(x - X_j(t)\right), \tag{3}$$

where $X_j(t), j = 1, \ldots, N$ represent the location of the N particles. We use capital X to indicate that the positions are random variables. The density of the particles determines the value of v, and heuristically one thinks of m_j as the "mass" of the jth particle.

We recover u from v by $u(x,t) = \int_{-\infty}^{x} v(x',t)\colon dx'$:

$$u(x,t) = \sum_{j=1}^{N} m_j H\left(x - X_j(t)\right). \tag{4}$$

Thus u is represented as a step function with jumps of size m_j at X_j. If all the particles have the same mass, m, the value of u at x is m times the number of particles that lie to the left of x. The boundary condition at $-\infty$ is automatically satisfied, and the condition at $+\infty$ is satisfied on average, with fluctuations [19]. This corresponds to conservation of mass.

As described in [5], there is much less noise in the computed value of u than of v because all the particles contribute to the value at any point, x. Moreover, if the particles have equal mass, their density is large precisely where u has large gradients.

Once the initial data is discretized, the GRW method evolves the particle positions and masses such that u satisfies equation (1). This is done by a fractional step iteration in which the diffusion term is modeled by a random walk and the reaction term is modeled as exponential growth-or decay of the particle masses. For each time step the sequence is:

A. Gaussian Random Walk Step:

$$X_j(t + \Delta t) = X_j(t) + \sigma_j$$

where the σ_j are independent $N(0, 2\nu\Delta t)$ random variables

B. Evaluate $u_j = u(X_j(t + \Delta t)), j = 1, \ldots, N$ using (4)

C. Kill or Replicate Particles: with probability $|f'(u_j)|\Delta t$
   i. Kill particle if $f' < 0$
   ii. Create a new particle at X_j if $f' > 0$

Note that the only way the particles interact with each other, and hence the only way that the nonlinearity of the equations is manifest in the algorithm, is that the local value of u depends on the positions of all the particles. Not surprisingly, this is the step that is most difficult computationally. Naively, computing equation (4) for all the X_j's requires $O(N^2)$ work, but if the particles are sorted, it requires only $O(N)$ work plus $O(N \log N)$ for the sorting.

The variant of this method due to Chorin and Ghoniem [1, 12, 5] is essentially the same, except that instead of killing and replicating particles, the masses are increased or decreased according to the ordinary differential equation

$$\frac{dm_j}{dt} = f'(u_j).$$

In the mean and to $O(\Delta t)$ the two procedures are equivalent.

Both methods have been used to solve traveling-wave problems in one-dimension. The mechanism of wave propagation is transfer of mass from behind the front to ahead of it. With randomized particle death and replication, particles are killed off behind the wave front, where $f'(\cdot)$ is mostly negative, and replicated ahead of the front, where $f'(\cdot)$ is mostly positive. With deterministic particle growth and decay, there is a graded transmission of mass from neighbor to neighbor.

3. The Deterministic Algorithm.

Among both the different Monte Carlo algorithms we discussed in the previous section, the Chorin-Ghoniem variant already contains deterministic elements. However, it requires

using time dependent masses for the particles. What we seek is a totally deterministic algorithm where the masses of the particles remain constant and equal throughout the computation. This kind of algorithm has been applied to convection-diffusion equations before [3, 6, 13, 15, 16, 17], and we wish to extend these methods to reaction-diffusion problems.

To begin our derivation, let us assume that we have a monotonic solution to (1), $u(x,t)$ with $0 \leq u(x,t) \leq 1$ for all $t > 0$ and $u(-\infty,t) = 0, u(+\infty,t) = 1$. Then we may define the functions $x_i(t)$ implicitly by saying they mark the locations of where the solution attains certain values as:

$$u(x_i(t),t) = \frac{i}{N}, \quad i = 1,2,\ldots,N-2,N-1, \quad t \geq 0. \tag{5}$$

Obviously we have that $x_0(t) = -\infty$ and $x_N(t) = +\infty$. We now seek equations of motion for the $x_i(\cdot)$'s given, that they satisfy (1). Let us take the total derivative with respect to time of (5):

$$\frac{d\ [u(x_i(t),t)]}{dt} = 0 = u_x(x_i(t),t)\dot{x}_i(t) + u_t(x_i(t),t). \tag{6}$$

A rearrangement of (6) gives:

$$\dot{x}_i(t) = \frac{-1}{u_x(x_i(t),t)} u_t(x_i(t),t) = \frac{-1}{u_x(x_i(t),t)} \Big(\nu u_{xx}(x_i(t),t) + f(u(x_i(t),t)) \Big). \tag{7}$$

Equation (7) gives an expression for the velocity of the ith marker point, $\dot{x}_i(t)$, in terms of certain derivatives of u and values of $f(u(\cdot))$. We have defined the values of u at the points $x_i(t)$, so let us use those values to evaluate the terms above. Let us denote $u(x_i(t),t)$ as u_i. Using formally second order finite-difference approximations we first approximate:

$$\frac{-1}{u_x(x_i(t),t)} \approx -\frac{x_{i+1}(t) - x_{i-1}(t)}{u_{i+1} - u_{i-1}} = -\frac{N}{2}\Big(x_{i+1} - x_{i-1} \Big). \tag{8}$$

Next we approximate:

$$u_{xx}(x_i(t),t) \approx \frac{\frac{u_{i+1}-u_i}{x_{i+1}(t)-x_i(t)} - \frac{u_i - u_{i-1}}{x_i(t) - x_{i-1}(t)}}{\left(\frac{x_{i+1}(t)+x_i(t)}{2}\right) - \left(\frac{x_i(t)+x_{i-1}(t)}{2}\right)} =$$

$$\frac{2}{N\,(x_{i+1} - x_{i-1})} \left[\frac{1}{x_{i+1} - x_i} - \frac{1}{x_i - x_{i-1}} \right] = \tag{9}$$

$$\left[-\frac{N}{2}\Big(x_{i+1} - x_{i-1} \Big) \right]^{-1} \left[\frac{1}{x_i - x_{i-1}} - \frac{1}{x_{i+1} - x_i} \right].$$

Finally it is obvious that

$$f(u(x_i(t),t)) = f\left(\frac{i}{N} \right), \tag{10}$$

so that the resulting equation becomes:

$$\dot{x}_i = \nu\left[\frac{1}{x_i - x_{i-1}} - \frac{1}{x_{i+1} - x_i}\right] - \frac{N}{2}\left(x_{i+1} - x_{i-1}\right) f\left(\frac{i}{N}\right). \tag{11}$$

This equation is very odd in the sense that its linear terms correspond to the nonlinear terms of the PDE (1) and visa versa. The first term on the right-hand side of (11) is highly nonlinear and corresponds to the linear diffusion term of (1), while the nonlinear $f(u)$ term of the PDE results in the second (linear) term of (11). This is not, however, the complete system to consider. Usually equation (1) is part of an initial-value or initial boundary-value problem. Thus we must consider ways of incorporating the various boundary conditions associated with these problems into this curious formulation.

4. Boundary Conditions.

The system of ODEs in (11) is quite contrary with regard to linear and nonlinear terms. It is also a contrarian with regard to boundary conditions since the most natural problem to discretize is the pure initial-value problem with a monotonic solution. The pure initial-value problem on the entire line presents difficulties for finite-difference and finite-element methods. In these cases either explicit methods are used and only that subset of the line containing the dynamic parts of the solution is included in the computation, or an artificial boundary is used. With this deterministic particle method no such *ad hoc* approach is required.

Another way that this discretization is contrary to finite-difference and finite-element methods is that here we perform a uniform discretization of the dependent variable, u. In finite-difference and finite-element methods it is space, an independent variable, that is discretized to give a system of ODEs. While this traditional approach makes certain types of error analysis more convenient, discretization of the solution permits computational element, the particles, to concentrate automatically in areas where the solution is changing most rapidly.

For the monotonic pure initial-value problem we have that $x_0(t) = -\infty$ and $x_N(t) = +\infty$ at all times, and so $\dot{x}_0(t) = \dot{x}_N(t) = 0$. For $i = 2, \ldots, N-2$ we can just apply equation (11). The difficulty arises for $i = 1$ and $i = N-1$ since equation (11) involves the points x_0 and x_N in the terms involving f. These difficulties can be avoided if we go back to the derivation for these two equations and rederive them, being mindful to avoid the singularities. We may think of the ODEs in (11) as a result of combining the ODEs derived from the particle method discretization of the two equations (a) $u_t = \nu u_{xx}$ and (b) $u_t = f(u)$ under the assumption of a monotonic solution. Discretization of (a) yields $\dot{x}_i = \nu\left[\frac{1}{x_i - x_{i-1}} - \frac{1}{x_{i+1} - x_i}\right]$, which is valid at $i = 1$ and $i = N-1$, as the terms involving the infinite quantities are zero. To rederive (b) let us use the relevant part of equation (7). This means consideration of $\dot{x}_i(t) = \frac{-f(u(x_i(t),t)}{u_x(x_i(t),t)}$. At $i = 1$ we may approximate $u_x(x_1(t),t) \approx \frac{u_2 - u_1}{x_2 - x_1} = \frac{1}{N(x_2 - x_1)}$, while at $n = N-1$ we have $u_x(x_{N-1}(t),t) \approx \frac{1}{N(x_{N-1} - x_{N-2})}$. These along with the contributions

from (a) give the following system for the pure initial-value problem:

$$
\begin{aligned}
\dot{x}_0 &= 0, \\
\dot{x}_1 &= -\nu\left[\frac{1}{x_2 - x_1}\right] - N\,(x_2 - x_1)\,f\left(\frac{1}{N}\right), \\
\dot{x}_i &= \nu\left[\frac{1}{x_i - x_{i-1}} - \frac{1}{x_{i+1} - x_i}\right] - \frac{N}{2}\,(x_{i+1} - x_{i-1})\,f\left(\frac{i}{N}\right), \quad i = 2, \ldots, N-2, \\
\dot{x}_{N-1} &= \nu\left[\frac{1}{x_{N-1} - x_{N-2}}\right] - N\,(x_{N-1} - x_{N-2})\,f\left(\frac{N-1}{N}\right), \\
\dot{x}_N &= 0.
\end{aligned} \tag{12}
$$

Similarly, Dirichlet boundary conditions are easy to impose with this particle method. Consider first a time independent Dirichlet boundary condition:

$$u(0,t) = U_0, \qquad u(1,t) = U_1, \qquad U_0 < U_1. \tag{13}$$

Here the solution is to rescale the problem so that $U_0 = 0$ and $U_1 = 1$ and use the conditions $x_0 = 0$ and $x_N = 1$. The situation is a bit more complicated when the Dirichlet boundary conditions are time-dependent:

$$u(0,t) = U_0(t), \qquad u(1,t) = U_1(t), \qquad U_0(t) < U_1(t), \quad t \geq 0. \tag{14}$$

Since we have discretized u in our particle method, time-dependent boundary conditions require the creation and destruction of particles along the boundary. For example, if $U_0(t)$ is increasing and in some time interval gains N^{-1} in value, the left-most particle's position must be set to zero. This has the effect of increasing the value of the solution at zero by N^{-1}. Conversely, if $U_0(t)$ is decreasing and diminishes by N^{-1} in value, a new particle is introduced at zero. At $x = 1$ we use a similar strategy, except that we remove a particle when $U_0(t)$ decreases and insert a particle when $U_0(t)$ increases.

In many cases Neumann boundary conditions are appropriate. The time-independent version of these conditions

$$u_x(0,t) = U_0, \qquad u_x(1,t) = U_1, \qquad U_0 \geq 0, \qquad U_1 \geq 0. \tag{15}$$

Since we are restricting our method to monotonic solutions, we must have $u_x(0,t) \geq 0$ and $u_x(1,t) \geq 0$. Let us first consider the leftmost boundary condition. Suppose that the left-most particle is at the location $x_1 > 0$. An estimate of the derivative is thus

$$u_x(0,t) \approx \frac{u_1 - u_0}{x_1} = \frac{1}{N x_1} = U_0 \rightarrow x_1 = \frac{1}{N U_0}. \tag{16}$$

Similarly, at the right-hand side assume that x_{N-1} is the right-most particle in the unit interval, then the right-hand Neumann condition becomes $x_{N-1} = 1 - \frac{1}{NU_1}$.

It should be clear that time-dependent Neumann boundary conditions are really no more trouble. Indeed, if we have the conditions:

$$u_x(0,t) = U_0(t), \qquad u_x(1,t) = U_1(t), \qquad U_0(t) \geq 0, \qquad U_1(t) \geq 0. \tag{17}$$

They are enforced via: $x_1(t) = \frac{1}{NU_0(t)}$ and $x_{N-1}(t) = 1 - \frac{1}{NU_1(t)}$. One important point is that we are constrained to keep these particles in the unit interval so that $1 > x_1(t) = \frac{1}{NU_0(t)}$ and $0 < x_{N-1}(t) = 1 - \frac{1}{NU_1(t)}$, which means that $N > \frac{1}{U_0(t)}$ and $N > \frac{1}{U_1(t)}$ for all time.

5. Time Discretizations.

Let us now consider some simple time discretizations of (11). We will consider only the forward- and backward-Euler methods here, which are first order methods. However, the analysis of the backward-Euler method will indicate the magnitude of the stiffness expected in these ODEs when solved by other implicit methods, [4].

Let us consider the pure initial-values problem that results in the system of ODEs in (12). Let us first give this equation the vector notation: $\dot{\mathbf{x}}(t) = F(\mathbf{x}(t))$, where $\mathbf{x}(t) = (x_0(t), x_1(t), \ldots, x_{N-1}(t), x_N(t))^T$ and $F(\mathbf{x}(t))$ is the right-hand side of (12). We now introduce a time discretization with time step Δt and $\mathbf{x}_j = \mathbf{x}(j\Delta t)$ so that the forward Euler method is merely:

$$\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t} = F(\mathbf{x}_n) \rightarrow \mathbf{x}_{n+1} = \mathbf{x}_n + \Delta t F(\mathbf{x}_n). \tag{18}$$

With our particular $F(\cdot)$, this equation is known to converge to the exact solution as $\Delta t \rightarrow 0$ as $O(\Delta t)$, [2]. In fact, it can be shown that if $\frac{\Delta t}{\min_i[(x_{i+1}-x_i)(x_i-x_{i-1})]} \leq \frac{1}{2\nu}$ this difference equation is numerically stable. This fact is very much analogous to the stability of the forward-Euler method for a finite-difference solution to these same equations,[8, 14].

Because of the serious time-step restriction, we should consider the fully implicit backward-Euler method:

$$\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t} = F(\mathbf{x}_{n+1}) \rightarrow \mathbf{x}_{n+1} - \Delta t F(\mathbf{x}_{n+1}) = \mathbf{x}_n. \tag{19}$$

This equation forms a nonlinear tridiagonal system. To see this let us consider the ith equation of equation (19):

$$x_i^{n+1} - \Delta t \nu \left[\frac{1}{x_i^{n+1} - x_{i-1}^{n+1}} - \frac{1}{x_{i+1}^{n+1} - x_i^{n+1}} \right] + \Delta t \frac{N}{2} \left(x_{i+1}^{n+1} - x_{i-1}^{n+1} \right) f\left(\frac{i}{N} \right) = x_i^n. \tag{20}$$

The second term corresponds to the linear diffusion term in (1), and it can be simplified by reduction to a common denominator. Let us define $K_i^{n+1} = \frac{\nu}{(x_i^{n+1} - x_{i-1}^{n+1})(x_{i+1}^{n+1} - x_i^{n+1})}$ so that equation (20) becomes:

$$\begin{aligned} l_i^{n+1} x_{i-1}^{n+1} + d_i^{n+1} x_i^{n+1} + u_i^{n+1} x_{i+1}^{n+1} &= x_i^n, \\ l_i^{n+1} &= -\Delta t \left[K_i^{n+1} + \frac{N}{2} f\left(\frac{i}{N} \right) \right], \\ d_i^{n+1} &= 1 + 2\Delta t K_i^{n+1}, \\ u_i^{n+1} &= -\Delta t \left[K_i^{n+1} - \frac{N}{2} f\left(\frac{i}{N} \right) \right]. \end{aligned} \tag{21}$$

Solving (21) requires dealing with a very odd nonlinear system. Even if the nonlinear terms from diffusion are held constant (the K_i^{n+1}'s), the linear system in (21) is not

guaranteed to be diagonally dominant.[1] However, a simple Picard iteration that will converge is possible if we rearrange (21) to allow us to work with a diagonally dominant system. Let us put the terms involving $f(\cdot)$ on the right-hand side as follows:

$$
\begin{aligned}
L_i^{n+1}x_{i-1}^{n+1} + D_i^{n+1}x_i^{n+1} + U_i^{n+1}x_{i+1}^{n+1} &= x_i^n + \Delta t\frac{N}{2}f\left(\frac{i}{N}\right)\left(x_{i+1}^{n+1} - x_{i-1}^{n+1}\right), \\
L_i^{n+1} &= -\Delta t K_i^{n+1}, \\
D_i^{n+1} &= 1 + 2\Delta t K_i^{n+1}, \\
U_i^{n+1} &= -\Delta t K_i^{n+1}.
\end{aligned} \tag{22}
$$

This nonlinear system is diagonally dominant, and so the Picard iteration defined as follows will converge, [9]:

A. Initialization: $\mathbf{x}^{old} = \mathbf{x}^n$

B. Iterate until converged:

i. Solve the following for $\mathbf{x}^{new}$:

$$
\begin{aligned}
L_i^{old} &= -\Delta t K_i^{old}, \\
D_i^{old} &= 1 + 2\Delta t K_i^{old}, \\
U_i^{old} &= -\Delta t K_i^{old}, \\
L^{old}x_{i-1}^{new} + D^{old}x_i^{new} + U^{old}x_{i+1}^{new} &= \\
x_i^n + \Delta t\frac{N}{2}f\left(\frac{i}{N}\right)&\left(x_{i+1}^{old} - x_{i-1}^{old}\right).
\end{aligned} \tag{23}
$$

ii. Set $\mathbf{x}^{old} = \mathbf{x}^{new}$

C. Form the solution to (22) as $\mathbf{x}^{n+1} = \mathbf{x}^{new}$.

Notice that the linear system in (23) is of the same form as one encounters in a finite-difference solution to (12). In fact, instead of the term $L_i^{old} = -\Delta t K_i^{old}$ we would have $\frac{-\nu\Delta t}{\Delta x^2}$ in the finite-difference case. These terms are very much analogous. Note that the denominator of $-\Delta t K_i^{old} = \frac{-\nu\Delta t}{(x_i^{old} - x_{i-1}^{old})(x_{i+1}^{old} - x_i^{old})}$ is the product of two neighboring variable spatial grid spacings, just like in the finite-difference case.

Now let us consider solving (21) with a Newton iteration. Recall that the Newton method iteration for the solution to the nonlinear system $\mathbf{G}(\mathbf{x}) = \mathbf{0}$ is defined implicitly by the equation $\mathbf{J}(\mathbf{x}_{old})(\mathbf{x}_{new} - \mathbf{x}_{old}) = -\mathbf{G}(\mathbf{x}_{old})$. Here the Jacobian matrix, $\mathbf{J}$, has elements defined by $[\mathbf{J}]_{ij} = \frac{\partial G_i}{\partial x_j}$. For equation (21) we have

$$
G_i(\mathbf{x}) = x_i - \nu\Delta t\left[\frac{1}{x_i - x_{i-1}} - \frac{1}{x_{i+1} - x_i}\right] + \Delta t\frac{N}{2}\left(x_{i+1} - x_{i-1}\right)f\left(\frac{i}{N}\right) - x_i^n, \tag{24}
$$

[1] Diagonal dominance guarantees both nonsingularity and the numerical stability of Gaussian elimination without pivoting, the Thomas algorithm. Without diagonal dominance we must pivot, which slows down the computation.

where the final, converged, solution gives $\mathbf{x}_{n+1}$. Thus $\mathbf{J}$ has the form:

$$[\mathbf{J}]_{ij} = \begin{cases} \frac{-\nu\Delta t}{(x_{i+1}-x_i)^2} + \frac{N\Delta t}{2} f\left(\frac{i}{N}\right), & j = i+1 \\ 1 + \left[\frac{\nu\Delta t}{(x_{i+1}-x_i)^2} + \frac{\nu\Delta t}{(x_i-x_{i-1})^2}\right], & j = i \\ \frac{-\nu\Delta t}{(x_i-x_{i-1})^2} - \frac{N\Delta t}{2} f\left(\frac{i}{N}\right), & j = i-1 \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

The matrix $\mathbf{J}$ is tridiagonal with a very curious form: it is the sum of a strictly diagonally dominant symmetric matrix and a real antisymmetric matrix. The result is a tridiagonal matrix that, in general, cannot be thought of as diagonally dominant. However, for most cases of interest this system will be diagonally dominant but not symmetric. Since we are adding and subtracting the same number from the off diagonals, the matrix will remain diagonally dominant if $[\mathbf{J}]_{ii+1} < 0$. This requirement is equivalent to $\frac{\nu\Delta t}{(x_{i+1}-x_i)^2} > \frac{N\Delta t}{2} f\left(\frac{i}{N}\right)$. Replace the term involving f with its maximum on the unit interval, $f_{max} = \max_{x\in[0,1]} f(x)$, and divide by $\Delta t N^2$ to give $\frac{\nu}{[N(x_{i+1}-x_i)]^2} > \frac{1}{2N} f_{max}$. Now the term $\frac{1}{[N(x_{i+1}-x_i)]^2}$ is an approximation to the square of the first derivative. Since the solution we seek is monotonic, this is always bounded away from zero by a constant, $H^{-1} > 0$, thus we rewrite this equation as $N > \frac{H f_{max}}{2\nu}$. Since the right-hand side is constant, choosing N larger makes the inequality true; thus by choosing N sufficiently large, the system can be made diagonally dominant.

6. Numerical Results.

In this section we consider the numerical solution of Nagumo's equation, [11], as part of a pure initial value problem:

$$u_t = u_{xx} + u(1-u)(u-a), \quad t > 0, \quad u(x,0) = u_0(x), \quad 1 \geq a \geq 0. \tag{26}$$

In this pure initial-value problem setting, Naugmo's equation has a stable traveling wave solution given by $u(x,t) = \frac{1}{1+e^{-(x-\theta t)/\sqrt{2}}}$, where the wave speed is $\theta = \sqrt{2}\left(a - \frac{1}{2}\right)$. When $a = \frac{1}{2}$ we have $\theta = 0$, and the solution is a stable standing wave. These properties are easy to check numerically and make Nagumo's equation a good candidate to explore the properties of a new numerical method for this class of problems.

We begin by computing numerically the solution to Nagumo's equation with our particle method using $u_0(x) = \frac{1}{1+e^{-x/\sqrt{2}}}$. This implies that $u(x,t) = \frac{1}{1+e^{-(x-\theta t)/\sqrt{2}}}$. In figure 1, below, we plot the computed solution at uniformly spaced times for the case where $a = 0.75$, with $N = 64$ and $\Delta t = 0.1$. Notice that the particles move with the solution tracking the steep gradient. In addition, by the end of the computations every single particle has moved past the position of the rightmost particle in the initial configuration. This shows how this method is naturally adapting to the solution and is equitable in its distribution of particles.

A qualitative observation that can be noticed by scrutinizing figure 1 is that the particles seem to be moving slightly away from the exact solution. This observation seems to indicate that actual wave speed of the particles may be slightly greater than θ, the wave speed of the exact solution. Fortunately, one calculation that we can carry out explicitly is the

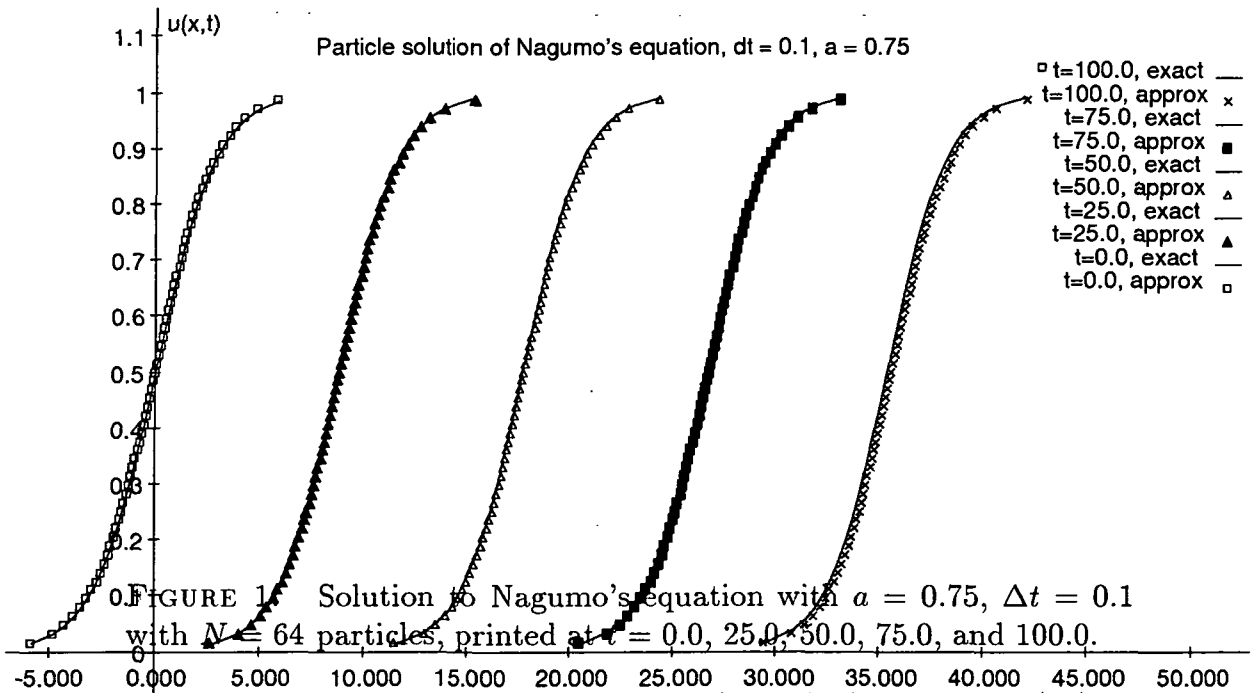

FIGURE 1. Solution to Nagumo's equation with $a = 0.75$, $\Delta t = 0.1$ with $N = 64$ particles, printed at $t = 0.0$, 25.0, 50.0, 75.0, and 100.0.

computation of the exact wave speed of the particle method solution to (26) to compare to the wave speed of the exact solution. Consider the wave speed of the midpoint of the travelling wave. Recall that the midpoint of the solution to (26) is the point, $x_{mid}(t)$, defined implicitly by $u(x_{mid}(t), t) = \frac{1}{2}$. By inspection we see that $x_{mid}(t) = \theta t$ in the exact solution. As a check, we note that the exact velocity of the midpoint is also given by applying equation (7) to the midpoint:

$$\dot{x}_{mid}(t) = \frac{-1}{u_x(x_{mid}(t), t)} \left[u_{xx}(x_{mid}(t), t) + \frac{1}{2}\left(1 - \frac{1}{2}\right)\left(\frac{1}{2} - a\right) \right]. \tag{27}$$

It is straightforward to calculate that $u_x(x_{mid}(t), t) = \frac{1}{4\sqrt{2}}$ and $u_{xx}(x_{mid}(t), t) = 0$. This gives $\dot{x}_{mid}(t) = -4\sqrt{2}\left[\frac{1}{4}\left(\frac{1}{2} - a\right)\right] = \sqrt{2}\left(a - \frac{1}{2}\right) = \theta$, as we expected.

Let us now calculate what the wave speed is of the midpoint when the exact solution is substituted into the discrete equations. We must therefore consider applying equation (11), with $i = mid$, $\nu = 1$, and $f(u) = u(1-u)(u-a)$, to $u(x,t) = \frac{1}{1+e^{-(x-\theta t)/\sqrt{2}}}$.

First we show that the diffusion term in (11) zero for Nagumo's equation. Assume we have N points in our discretization, where N is even. Hence $mid = \frac{N}{2}$. We also have that $\frac{mid+1}{N} = \frac{1}{1+e^{-(x_{mid+1}(t)-\theta t)/\sqrt{2}}}$, which can be solved to give $x_{mid+1}(t) = \theta t - \sqrt{2}\ln(\frac{N}{mid+1} - 1)$. Similarly, $x_{mid-1}(t) = \theta t - \sqrt{2}\ln(\frac{N}{mid-1} - 1)$, so that the differences in the diffusion term are given by:

$$
\begin{aligned}
x_{mid+1} - x_{mid} &= -\sqrt{2}\left[\ln\left(\frac{N}{mid+1} - 1\right) - \ln\left(\frac{N}{mid} - 1\right)\right] = \\
&\quad -\sqrt{2}\ln\left(\frac{\frac{N}{mid+1} - 1}{\frac{N}{mid} - 1}\right), \\
x_{mid} - x_{mid-1} &= -\sqrt{2}\left[\ln\left(\frac{N}{mid} - 1\right) - \ln\left(\frac{N}{mid-1} - 1\right)\right] = \\
&\quad -\sqrt{2}\ln\left(\frac{\frac{N}{mid} - 1}{\frac{N}{mid-1} - 1}\right).
\end{aligned} \tag{28}
$$

Now we compute:

$$
(x_{mid+1} - x_{mid}) - (x_{mid} - x_{mid-1}) = -\sqrt{2}\ln\left[\left(\frac{\frac{N}{mid+1} - 1}{\frac{N}{mid} - 1}\right)\left(\frac{\frac{N}{mid-1} - 1}{\frac{N}{mid} - 1}\right)\right]. \tag{29}
$$

The numerator of the term inside the logarithm can be rewritten as:

$$
\begin{aligned}
&\left(\frac{N}{mid+1} - 1\right)\left(\frac{N}{mid-1} - 1\right) = \frac{N^2}{(mid+1)(mid-1)} - \frac{N}{mid-1} - \frac{N}{mid+1} + 1 = \\
&\frac{N^2 - N\left[(mid+1) + (mid-1)\right]}{(mid+1)(mid-1)} + 1 = \frac{N^2 - N\left[2(mid)\right]}{(mid+1)(mid-1)} + 1 = \\
&\frac{N^2 - N\left[2\left(\frac{N}{2}\right)\right]}{(mid+1)(mid-1)} + 1 = 1.
\end{aligned} \tag{30}
$$

And thus $(x_{mid+1} - x_{mid}) = (x_{mid} - x_{mid-1})$, thus making the diffusion term in (11) vanish. For convenience, in the sequel we will refer to the common value of this difference as h.

Thus we are left to consider the equation

$$
\dot{x}_{mid}(t) = -\frac{N}{2}(x_{mid+1} - x_{mid-1}) f\left(\frac{1}{2}\right) = -\frac{N}{2} 2h f\left(\frac{1}{2}\right). \tag{31}
$$

The term $\frac{N}{2}2h$ is the reciprocal of the centered difference approximation to the first derivative. Indeed it is the case that

$$
\begin{aligned}
&\frac{\frac{2}{N}}{2h} = \frac{u(x_{mid}+h,t) - u(x_{mid}-h,t)}{2h} = u_x(x_{mid},t) + \frac{h^2}{6}u_{xxx}(x_{mid},t) + O(h^4) \rightarrow \\
&\frac{N}{2}2h = \frac{1}{u_x(x_{mid},t) + \frac{h^2}{6}u_{xxx}(x_{mid},t) + O(h^4)} = \frac{1}{u_x(x_{mid},t)}\left[1 - \frac{h^2}{6}\frac{u_{xxx}(x_{mid},t)}{u_x(x_{mid},t)} + O(h^4)\right].
\end{aligned} \tag{32}
$$

For the exact solution one can show that $u_{xxx}(x_{mid},t) = -\frac{1}{16\sqrt{2}}$. Recall that above we stated that $u_x(x_{mid}(t),t) = \frac{1}{4\sqrt{2}}$ so that $\frac{u_{xxx}(x_{mid},t)}{u_x(x_{mid},t)} = -\frac{1}{4}$. Substituting this into (31) gives:

$$\dot{x}_{mid}(t) = \frac{-f\left(\frac{1}{2}\right)}{u_x(x_{mid}(t),t)}\left[1+\frac{h^2}{24}+O(h^4)\right] = \theta\left[1+\frac{h^2}{24}+O(h^4)\right]. \tag{33}$$

Equation (33) implies that the computed solution should have (1) a velocity that is slightly greater than the actual solution, and (2) this difference in velocity goes to zero with h^2. Intuition suggests that $h \propto \frac{1}{N}$, so one can think of the error in the velocity as going to zero with the square of the number of particles. We notice both these phenomena in our computations.

	a						
N	0.125	0.25	0.375	0.5	0.625	0.75	0.875
4	-1.71e-01	-1.33e-01	-7.32e-02	0.0e+00	7.32e-02	1.33e-01	1.71e-01
8	-1.04e-01	-7.36e-02	-3.77e-02	0.0e+00	3.77e-02	7.36e-02	1.04e-01
16	-4.70e-02	-2.68e-02	-1.18e-02	0.0e+00	1.18e-02	2.68e-02	4.70e-02
32	-1.91e-02	-8.78e-03	-3.23e-03	0.0e+00	3.23e-03	8.78e-03	1.91e-02
64	-7.62e-03	-2.80e-03	-8.23e-04	0.0e+00	8.23e-04	2.80e-03	7.62e-03
128	-3.06e-03	-9.00e-04	-1.96e-04	0.0e+00	1.96e-04	9.00e-04	3.06e-03
256	-1.25e-03	-2.91e-04	-4.35e-05	0.0e+00	4.35e-05	2.91e-04	1.25e-03
512	-5.14e-04	-9.58e-05	-8.72e-06	0.0e+00	8.72e-06	9.58e-05	5.14e-04
1024	-2.13e-04	-3.19e-05	-1.43e-06	0.0e+00	1.43e-06	3.19e-05	2.13e-04
2048	-8.88e-05	-1.07e-05	-1.15e-07	0.0e+00	1.15e-07	1.07e-05	8.88e-05
4096	-3.71e-05	-3.68e-06	4.73e-08	0.0e+00	-4.73e-08	3.68e-06	3.71e-05

TABLE 1. The error in the wave speed (θ) in the particle method solution to Nagumo's equation. Errors for various values of a and N are presented.

To confirm this calculation we have computed the velocity of $x_mid(t)$ for various values of a, and hence various values of θ, and with various (power-of-two) numbers of points. We have tabulated our results in table 1. A logarithmic regression actually gives the error in the wave speed as $O(N^{-\frac{3}{2}})$ instead of the expected $O(N^{-2})$.

We have now computed the systematic error in the wave speed of our numerical method. The overall error in our numerical method is the sum of this error, which should be $O(t)$, and the error in the solution is due only to the spatial discretization. Nagumo's equation allows us to study this second type of error directly by examining the numerical solution when $a = \frac{1}{2}$ and $\theta = 0$. In this case we expect that the numerical solution will deviate from the exact solution only, due to what is "discretization" error for this peculiar numerical

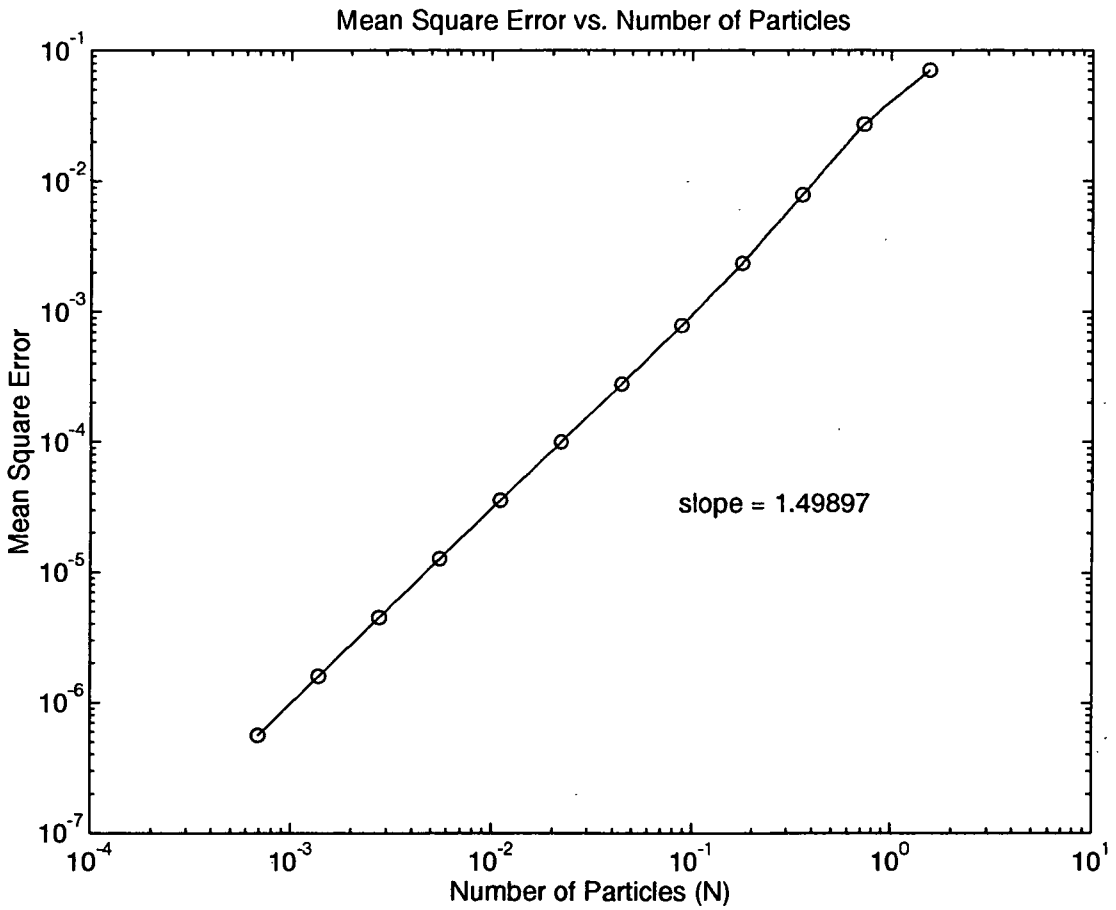

FIGURE 2. Mean square error of the particle method solution for various numbers of particles. The solution was computed with $a = \frac{1}{2}$ giving a wave speed of $\theta = 0$.

method as the solution is a standing wave. In fact, we see in table 1 that in this case there is no error in the computed wave speed for all values of h. In figure 2 we plot the

asymptotic mean square error in the particle method solution versus the number of points used. Notice that the slope of the curve in figure 2, a log-log plot, is ≈ -1.5, meaning that the the error behaves as $O(N^{-\frac{3}{2}})$. Thus the overall error of our method is $O(N^{-\frac{3}{2}})$.

7. Conclusions.

We have derived an adaptive particle method for the solution of one-dimensional reaction-diffusion equations with monotonic solutions. This method is the deterministic analog of a Monte Carlo method for these same problems. Like the Monte Carlo method, this deterministic method uses a discretization of the solution to obtain a method that naturally tracks the solution by resolving the gradient. Thus this method is particularly suited to equations which have very steep gradients in the solution. We have also analyzed the difference equations associated with this discretization and have proposed a Picard and Newton iteration for the solution of the nonlinear system of equations arising from the fully implicit backward Euler time discretization. We have also decomposed the error in the method into the error in computing the wave speed and the spatial discretization error. This former error can be asymptotically computed, while the later has been studied numerically. The overall error in the method appears to be $O(N^{-\frac{3}{2}})$. This is much better than the $O(N^{-\frac{1}{2}})$ error in the Monte Carlo method.

Many questions remain. The most important question being why is the accuracy $O(N^{-\frac{3}{2}})$ when we have chosen a spatial discretization based on second order accuracy? Perhaps the difference equations themselves can be refined to give a more accurate formulation. Or, the method of imposing the boundary conditions was only first order. This may have contaminated the overall accuracy of the solution. Is there a better formulation for the boundary conditions? It is also important to consider how to extend this method. Three ways that one would like to extend this method is to include (i) nonmonotonic but positive solutions, (ii) multi-species systems, and (iii) several spatial dimensions.

References

1. A. J. Chorin, *Numerical methods for use in combustion modeling*, Computing Methods in Applied Sciences and Engineering (R. Glowinski and J. L. Lions, eds.), North Holland, Amsterdam, 1980, pp. 229–235.
2. E. A. Coddington and N. I. Levinson, *Theory of Ordinary Differential Equations*, McGraw-Hill, New York, 1955.
3. P. Degond and F.-J. Mustieles, *A deterministic approximation of diffusion equations using particles*, SIAM. J. Sci. Stat. Comput. **11** (1990), 293–310.
4. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Engelwood Cliffs, New Jersey, 1971.
5. A. F. Ghoniem and F. Sherman, *Grid-free simulation of diffusion using random walk methods*, J. Comput. Phys. **61** (1985), 1–37.
6. F. Hermeline, *A deterministic particle method for transport diffusion equations: Application to the Fokker-Planck equation*, J. Comput. Phys. **82** (1989), 122–146.
7. A. L. Hodgkin and A. F. Huxley, *A quantitative description of membrane current and its application to conduction in the giant axon of Loligo*, J. Physiol. **117** (1952), 500–544.
8. E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, John Wiley and Sons, New York, 1966.
9. M. Mascagni, *The backward Euler method for the numerical solution of the Hodgkin-Huxley equations of nerve conduction*, SIAM J. Num. Anal. **27** (1990), 941–962.
10. H. P. McKean, Jr., *Application of Brownian motion to the equation of Kolmogorov-Petrovskii-Piskunov*, Communications on Pure and Applied Mathematics **XXVIII** (1975), 323–331.
11. J. Nagumo and S. Arimoto and S. Yoshizawa, *An active pulse transmission line simulating nerve axon*, Proc. IRE **50** (1962), 2061–2070.

12. A. K. Oppenheim and A. Ghoniem, *Application of the random element method to one-dimensional flame propagation problems*, AIAA-83-0600, AIAA 21st Aerospace Sciences Meeting, Reno, NV, 1983.
13. P. A. Raviart, *An analysis of particle methods*, Numerical methods in fluid dynamics (F. Brezzi, ed.), Lecture Notes in Mathematics, 1127, Springer-Verlag, New York, 1985.
14. R. D. Richtmeyer and K. W. Morton, *Difference Methods for Initial Value Problems*, Second Edition, Interscience Publishers, division of John Wiley and Sons, New York, London, Sydney, 1967.
15. G. Russo, *A particle method for collisional kinetic equations. I Basic theory and one-dimensional results*, J. Comput. Phys. **87** (1990), 270–300.
16. G. Russo, *Deterministic diffusion of particles*, Communications on Pure and Applied Mathematics **XLIII** (1990), 697–733.
17. G. Russo, *A Lagrangian method for collisional kinetic equations*, Proceedings of the 1988 AMS SIAM Summer Seminar, Colorado State University, Fort Collins, Colorado, July 18–29, 1988 (E. L. Allgower and K. Georg, eds.), Lectures in Applied Mathematics, 26, American Mathematical Society, Providence, Rhode Island, 1990, pp. 519–539.
18. A. S. Sherman and M. Mascagni, *A gradient random walk method for two-dimensional reaction-diffusion equations*, SIAM Journal on Scientific Computing **15** (1994), 1280–1293.
19. A. S. Sherman and C. S. Peskin, *A Monte-Carlo method for scalar reaction diffusion equations*, SIAM J. Sci. Stat. Comput. **7** (1986), 1360–1372.
20. A. S. Sherman and C. S. Peskin, *Solving the Hodgkin-Huxley equations by a random walk method*, SIAM J. Sci. Stat. Comput. **9** (1988), 170–190.
21. D. Talay, *Simulation and numerical analysis of stochastic differential systems: A review inbook Effective Stochastic Analysis* (1990), Inria (Report #1313), Valbonne, France, 1–51.

RIACS; NASA AMES RESEARCH CENTER; M/S T20G-5; MOFFETT FIELD, CALIFORNIA 94035-1000; CENTER FOR COMPUTING SCIENCES, I.D.A.; 17100 SCIENCE DRIVE; BOWIE, MARYLAND 20715-4300
E-mail address: mascagni@riacs.edu or mascagni@super.org

RIACS
Mail Stop T041-5
NASA Ames Research Center
Moffett Field, CA 94035